

## WEST Search History





DATE: Monday, February 23, 2004

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L10	19991291	19
<input type="checkbox"/>	L9	proxy near8 (cache or caching or cached) near8 (pay or paying or bill or billing or biling or restriction or policy or rule or regulation)	50
<input type="checkbox"/>	L8	19991291	7
<input type="checkbox"/>	L7	(prefetch or prefetching or prefetcher or prefetched) near8 (content or page or document or web) near8 (pay or paying or bill or billing or biling or restriction or policy or rule)	14
<input type="checkbox"/>	L6	19991291	14
<input type="checkbox"/>	L5	(prefetch or prefetching or prefetcher or prefetched) near8 (content or page or document or web) near8 link	32
<input type="checkbox"/>	L4	L3 and (prefetch or prefetching or prefetched)	1
<input type="checkbox"/>	L3	19991291	46
<input type="checkbox"/>	L2	((divide or divided or dividing) near8 content) same (portable or mobile)	137
<input type="checkbox"/>	L1	((divide or divided or dividing) near8 content) same (prefetch or prefetching or prefetched)	6

END OF SEARCH HISTORY

[First Hit](#)   [Fwd Refs](#)

Generate Collection

L6: Entry 1 of 14

File: USPT

Dec 16, 2003

DOCUMENT-IDENTIFIER: US 6665838 B1

TITLE: Web page thumbnails and user configured complementary information provided from a server

Application Filing Date (1):  
19990730Detailed Description Text (29):

Turning now to FIG. 9, there is shown a flowchart illustrating a basic overview for modifying the appearance of a web page in accordance with user selected criteria. A server receives a request for web content from a user along with user selected criteria (step 905) and then retrieves the requested web content (step 910). The server then parses the web page for user specified criteria (step 920). The web pages associated with any links found on the retrieved web page are prefetched and the server then parses the prefetched web pages for user specified criteria (step 930). If the prefetch fetches web pages that are several links removed from the presently viewed web page, the navigation assistant can parse these pages for user specified criteria as well. If the server finds some or all of the user specified criteria in the current web page or in one of the prefetched web pages, the current web page's appearance is changed in a user configured manner (step 950). Ideally, the change in the appearance would effect the representation of the link or the data found in a user customizable way. This may be actual modification of the data (in the case of HTML, the HTML source file which is downloaded to the browser would be modified to insert such things as tags (<b></b>(bold) or <i></i>(italics)), new replacement of images, etc. On the other hand it might also be done in a way that did not effect the raw data (e.g., the HTML source), but only the appearance of the unmodified data with regard to the viewer/browser. Either way this may mean inserting an icon at the presence of the data (e.g., an icon of a small eye as the character before a word or picture which had the preferred data as an alternate reference), or it may mean changing text color, text size, changing the font (e.g., making it bold or italic or both) or adding a translucent shading over an image or text, causing the text or image to appear selected (e.g., the same shading that occurs when you use a "find in document command"--typically inverting the text) or even adding a border or colored border around a word or image. It may also mean replacing or overlaying the data with something the user configured such as a "Do Not Enter Sign" or drawing a box around it.

Detailed Description Text (32):

After parsing the web page and configuring display settings in response to the presence or absence of user specified criteria, the navigation assistant parses the web page to determine if it contains links (universal resource locators "URLs") to other web pages (step 1135). If it does not, then the navigation assistant waits until a new web page is loaded at which point the process is repeated (step 1195). If the web page does contain links to other web pages (step 1135), then the server prefetches these other links (step 1150). If no the web page does not contain links to other web pages (step 1135), then the web page (with modifications, if any) is sent to the user (step 1193) and then the server waits until the user requests a new web page (step 1195).

Detailed Description Text (36):

A more detailed understanding of the steps involved in modifying the presentation of the web page (steps 1122-1130 and steps 1160-1180 of FIG. 11) is given with reference to FIG. 12. The server must first determine whether the user has configured the preferences for a layered prefetch (step 1205). In a layered prefetch, the navigator not only prefetches the web pages associated with the links on the currently displayed web page (first layer), but also prefetches the web pages associated with links on the web pages associated with the links on the currently displayed web page (second layer). This layered prefetch can be carried out to multiple layers (greater than two) as far as the user wishes consistent hardware and software constraints such as available memory and download speed.

Detailed Description Text (37):

If the user has configured the preferences for a layered prefetch, the server determines whether each linked web page contains an excessive number of broken links (step 1210). What constitutes an excessive number is determined by the user or plugin maker and can be given in terms of absolute numbers or in percentage of broken links to good links. A broken link is a link that, if followed, does not retrieve a web page or retrieves a web page containing a message indicating that the subject matter has moved, been removed, etc. If the navigator determines that the linked web page contains an excessive number of broken links, then the display is altered in some manner to indicate that the web page associated with that particular link contains an excessive number of broken links (step 1215). As an example, this could be accomplished by modifying the color of the link to perhaps gray where gray would be an indication to the user that the link contains an excessive number of broken links. Another example would be to display an image next to the link where the image would indicate that the web page associated with the link contains an excessive number of broken links. However, each of these is merely an example of a method of indicating the information to the user and, therefore, do not limit the scope of the present invention. Additionally, the server could parse second, third, etc. level pages for excessive broken links, etc.

Detailed Description Text (38):

If the user has not configured for a layered prefetch (step 1205) or if the web page associated with the link does not contain an excessive number of broken links (step 1210), then the navigator parses the present web page or prefetched web page, as the case may be, to determine whether it contains user configured negative preferences (step 1220). Negative preferences may be content related where the user has indicated key words or subject matter which is not wanted such as adult oriented material. Other examples of negative preferences include or relate to the size of the web page; avi's; music; number of links; number of images; total size of images; JavaScript presence; Java Applet presence; domain name suffix; author; and date of information, i.e. less than seven days old. If such unwanted material or characteristics are present on the web page, then the appearance of the currently viewed web page is altered to reflect such information (step 1225). Examples of such modification include presenting an image of a circle with a line through it next to the link to indicate that the associated web page contains unwanted characteristics.

[First Hit](#)   [Fwd Refs](#)☐ **Generate Collection**

L6: Entry 3 of 14

File: USPT

May 7, 2002

DOCUMENT-IDENTIFIER: US 6385641 B1

**\*\* See image for Certificate of Correction \*\***

TITLE: Adaptive prefetching for computer network and web browsing with a graphic user interface

Application Filing Date (1):19980605Brief Summary Text (7):

If we knew exactly which files a user needed next, we would retrieve only those files in advance. Assuming that the prefetch operation always finishes before the user requests the next file, then we would enjoy zero latency with no extra bandwidth consumption. Unfortunately, in reality, some prefetched information may never be used resulting in wasted bandwidth and increased delay to normal (nonprefetch) requests. This indicates that the prefetch problem has two aspects. The first one is how to estimate the probability of each file being accessed in the near future. In general, this probability changes with time as the user issues new requests. The second aspect is how to determine which files to prefetch so that the overall system performance can be optimized relative to some criteria. Previous studies on Internet prefetch focus only on the first aspect of the problem and either use a fixed threshold or prefetch a fixed number of the most popular links on a page [2, 7, 8] In the prior art PeakJet prefetching software for web browsers all available links on a page being viewed are prefetched [9]. If everyone were to employ this method, there would be unacceptable surges in network traffic. In fact, prefetching without caution could be disastrous to the network.

Detailed Description Text (118):

FIG. 13 shows the relation between the prefetch threshold and the average number of files prefetched for each user request, which indicates the amount of traffic increase in the entire system due to prefetching. When only the access probabilities from the client was used, this average number is very low, since the prediction was very accurate, and if the user had not visited the page at least 5 times, no link on it was prefetched. More files were prefetched for the other two cases, but still the average number of files prefetched for each user request is low. Comparing FIGS. 11 and 12 with FIG. 13, we can see that for a hit rate of more than 60%, the average number of files prefetched for each user request is less than two.

[First Hit](#)   [Fwd Refs](#)

Generate Collection

L6: Entry 5 of 14

File: USPT

Jan 8, 2002

DOCUMENT-IDENTIFIER: US 6338096 B1

TITLE: System uses kernals of micro web server for supporting HTML web browser in providing HTML data format and HTTP protocol from variety of data sources

Abstract Text (1):

The present invention provides a method and apparatus for transparently accessing multiple local and remote data stream types from an HTML browser. A micro web browser is disclosed which runs on the same platform as any commercial HTML browser, but in a separate process. Capabilities of the micro web browser include processing a data stream before the stream reaches the browser, translating different protocols to HTTP and different data types to HTML, where the translation is performed either on the fly or as pages are stored on the disk. The micro web server is further capable of allowing mixed media searches (e.g. searching first a users hard drive, then a users CD, then the World Wide Web. Additional capabilities of the micro web server include password protection, automated background download, intelligent prefetch of links, smart background printing, security mechanisms including page lockout and monitoring, decompression of media including CD ROMs and DVD, data stream modification including the addition of advertisements into streams and consistent buttons to all HTML pages.

Application Filing Date (1):

19980610

Detailed Description Text (13):

3) Kernal 3--Site/Page Background Fetch 58--a users perception of network speed can be enhanced by analyzing the page a user is currently accessing and prefetching links associated with that page in the background. This prefetching can be preempted at any time by a user request to fetch a specific page.

First Hit   Fwd Refs

☐ **Generate Collection**

L6: Entry 6 of 14

File: USPT

Aug 28, 2001

DOCUMENT-IDENTIFIER: US 6282542 B1

TITLE: Distributed system and method for prefetching objects

Application Filing Date (1):

19980805

CLAIMS:

8. The method of claim 1, wherein the document includes a hypertextual link to a second document, and the method further comprises prefetching the second document on the server side and forwarding the second document to the client side.

33. A system for providing Internet access via satellite, comprising:

an access point coupled to a plurality of web browsers; and

a satellite gateway coupled to the access point by a wireless satellite link, wherein the satellite gateway is remote from the web browsers and the access point, and is connected to the Internet such that the access point, the wireless satellite link, and the satellite gateway collectively provide a communications path between the web browsers and the Internet;

wherein the satellite gateway parses parent files of web pages requested by the web browsers to identify references to inline objects of such web pages, and in response to detection of a reference to an inline object, prefetches the inline object and forwards the inline object over the satellite link to the access point for delivery to a web browser.

40. In an Internet access system comprising a wireless satellite link having a client side and a server side which is remote from the client side, a method of reducing a delay associated with the retrieval of a web page which contains an object, the method comprising, on the server side of the satellite link:

retrieving a parent file of the web page from a remote web server in response to a request from a browser on the client side of the satellite link;

parsing the parent file of the web page to identify a reference to the object; and

in response to detection of the reference, prefetching the object from a web server, and forwarding the object over the satellite link to the client side for delivery to the browser;

whereby a need for the browser to retrieve the object over the satellite link following receipt of the parent file is avoided.

[First Hit](#)   [Fwd Refs](#)

Generate Collection

L6: Entry 7 of 14

File: USPT

May 8, 2001

DOCUMENT-IDENTIFIER: US 6230168 B1

TITLE: Method for automatically constructing contexts in a hypertext collection

Application Filing Date (1):19971126Detailed Description Text (72):

Transmission of information between a remote server 12 and client computers 10 can be optimized by prefetching the hypertext links on a requested web page and transmitting their contents to the client computers 10 that are viewing the web page. When a user takes a link to a web page that has been prefetched, the web page already resides in a cache 84 on the client computer 10 and therefore, the browser 82 only needs to render it rather than to fetch it and render it. This reduces the time required to satisfy and render the browser's request for the linked web page.

[First Hit](#)   [Fwd Refs](#)

Generate Collection

L6: Entry 8 of 14

File: USPT

Aug 1, 2000

DOCUMENT-IDENTIFIER: US 6098064 A

TITLE: Prefetching and caching documents according to probability ranked need S  
listApplication Filing Date (1):19980522Brief Summary Text (6):

In order to speed up access to a document that is selected by a user following a link, client computers can prefetch and cache documents. The number of documents that can be prefetched and cached depends on the amount of available cache on a client or server computer. Generally, the available resources on a client or server computer is small compared with the number of documents available on the web. In other words, only a small fraction of the expansive number of documents available on the web can be cached locally on a client or server computer. Consequently, the better a client or a server computer is able to identify documents that are most likely to be needed by a user, the better performance that user will experience while following linked documents or simply retrieving document from the web.



[First Hit](#)   [Fwd Refs](#)

Generate Collection

L6: Entry 9 of 14

File: USPT

Apr 25, 2000

DOCUMENT-IDENTIFIER: US 6055572 A

TITLE: System and method for creating pathfiles for use to predict patterns of web surfaces

Abstract Text (1):

A prefetching and control system for a computer network environment. The user configures the client's prefetch parameters which are: enabling/disabling prefetching, prefetch threshold value, and the maximum number of documents to prefetch. A prefetch value or weight is contained in the Hypertext Markup Language (HTML) page or prefetch file, called a pathfile, for each link. The HTML page contains the prefetch values for each of its links, while pathfile contains the weights for every link on the HTML page associated with the Universal Resource Locator (URL). The client compares the prefetch or weight values of each link with its threshold value to decide if the link should be prefetched and placed in the local cache as long as the maximum number of documents to prefetch is not exceeded. Pathfiles reside on the server and are created by the server or web administrator/author. The server automatically creates the pathfiles from its log files which are filtered to retain all of the valid document requests and average paths are derived from the filtered results. Weights are assigned to each path in the URL by the server and inserted into the pathfile along with the associated paths. If no log files exist on the server, then the web administrator/author may manually enter in the weights for each path.

Application Filing Date (1):

19980120

Brief Summary Text (8):

Many Internet browsers and add-on programs currently perform a "blind" prefetch, whereupon the browser pulls down all of the document links on a web page before the user has actually clicked on them. For example, in a very simple model of blind prefetching, the client pulls down page A from a server, parses the document links on that page and pulls each of them into the local cache. When the user clicks on a link to page B, it is loaded from the prefetched copy in the cache.

Brief Summary Text (10):

It would be advantageous to provide a prefetching and control system that improves the user-perceived network performance by allowing the client to efficiently decide what documents to prefetch, thereby reducing the demand on the server and network. It would further be advantageous to provide a prefetching and control system that enables the server to designate to the client what document links are important or the most common on a page.

Brief Summary Text (13):

In a preferred embodiment of the invention, the user configures the client's prefetch parameters which are: enabling/disabling prefetching, prefetch threshold value, and the maximum number of documents to prefetch. A prefetch value or weight is contained in the Hypertext Markup Language (HTML) page or prefetch file, called a pathfile, for each link. The HTML page contains the prefetch values for each of its links, while pathfile contains the weights for every link on the HTML page associated with the Universal Resource Locator (URL). The client compares the

prefetch or weight values of each link with its threshold value to decide if the link should be prefetched. If the value of the link is greater than the threshold value, then the link is prefetched and placed in the local cache as long as the maximum number of documents to prefetch is not exceeded. The client efficiently uses its free bandwidth by prefetching the documents that are most likely to be read by the user, thereby improving the user-perceived network performance and reducing the demand on the server and network.

Detailed Description Text (3):

Referring to FIG. 1, a browser retrieves a page 101. During a blind prefetch, the browser requests the documents for all of the links on the page 101. Here, the browser requests the documents for the Developers 102, Learning Site 103, Product Training 104, AppFoundry 105, Tech Support 106, and Download 107 links from the server and places them in the local cache.

Detailed Description Text (7):

The client 201 uses the prefetch values 308 in the HTML document 306 or pathfile 309 to decide which links on the page are the most likely to be accessed by a typical user. It prefetches the number of documents that it has been configured for from the server 203 according to the threshold value that the user configured on the client 201. Documents are prefetched that have prefetch values above the user-configured threshold value. The HTML document 306 contains prefetch values 308 for each document link contained on that page, while the pathfile 309 contains the prefetch values for every link in the URL.

Detailed Description Text (8):

A prefetching client first attempts to parse the contents and extract the prefetch information from the ANCHOR and the LINK (described later in this text) tags. The ANCHOR tag appears in the HTML document.

Detailed Description Text (19):

The client has a very simple design because it interacts and uses the functionality of already existing modules. It reads the preferences for the user and, if prefetching is turned on, it builds a list of documents to prefetch from the current page based on the ANCHOR tag's PRE value. The client skips those links which have a PRE value less than the one specified by the user. It then removes those links which are already in the cache or currently being downloaded (possibly in a GETLIST). The client then proceeds to download these documents on a lower priority. The speed of the network connection is a useful factor in determining what and how much to prefetch. Links that have been prefetched may be displayed in a different color to convey to the user which documents that have been prefetched.

Detailed Description Text (28):

To assist a prefetching client, the server creates pathfiles for each resident URL. When the client sends a GET request for the document, the server replies with an additional LINK header referencing "prefetching" and an HREF pointing to the location of the pathfile.

Detailed Description Text (94):

If the client is unable to parse the document for prefetch values, then it checks for a pathfile path in the LINK tag 403. If a path exists, the client checks if the pathfile is already in the cache 601. If it is not, then the client will get the pathfile from the server 602. The client parses the pathfile and creates the path that has the highest prefetch value above the threshold value 603. If the document prefetch limit has been reached 604, the prefetch process ends 605. Otherwise, the next link in the path is selected 606. If the link does not exist 607, then it is the path has ended and the prefetch process ends 605. If there is a link, then the link is checked to see if it is already in the cache 606. If so, then the next iteration of the loop is performed 604. If the link is not in the cache, then the client retrieves the document from the server 609 and places the document in the

cache 610. The loop then continues 604.

Detailed Description Text (104):

Predictive prefetching based on keyword indices. Users have a tendency to visit pages that have a common theme. The client queries each link on the page for keywords and follows a keyword pattern, e.g. sport, for document prefetching, giving higher weight to the links with the appropriate keyword matches.

Detailed Description Text (106):

Hub and spoke patterns. Hub and Spoke patterns occur when there is a main page that is the central hub for all of the links, for example, the user returns to the main page each time a link on the page is visited so he can follow other links on the main page. In this case, the prefetch would be weighted toward all of the links on the main page.

CLAIMS:

1. A process for generating pathfiles in a computer network environment, comprising the step of:

creating a pathfile from the server log files;

wherein said pathfile is a prefetch file that contains paths within the HTML page associated with the Universal Resource Locator (URL) and their associated prefetch values; and

wherein said prefetch values determine the relative weight of each link in said document.

9. An apparatus for generating pathfiles in a computer network environment, comprising:

a module for creating a pathfile from the server log files;

wherein said pathfile is a prefetch file that contains paths within a Universal Resource Locator (URL) and their associated prefetch values; and

wherein said prefetch values determine the relative weight of each link in said document.

[First Hit](#)   [Fwd Refs](#)☐ **Generate Collection**

L6: Entry 10 of 14

File: USPT

Feb 15, 2000

DOCUMENT-IDENTIFIER: US 6026474 A

TITLE: Shared client-side web caching using globally addressable memory

Application Filing Date (1):19970502Brief Summary Text (9):

Another attempt to optimize Web access involves prefetching Web pages. For example, when a user downloads a Web page, the system may also download Web pages which are linked to that page. Should the user traverse a link to one of the prefetched Web pages, the system is able to display the new page to the user with little or no delay. These systems, however, do not allow users to share Web caches and have an additional drawback that network traffic is increased. For example, these systems must prefetch multiple pages to provide an end-user with quick access to only one page, and the user may select a page that is not linked to the previous page viewed by the user that, therefore, has not been prefetched.

First Hit    Fwd Refs☐ **Generate Collection**

L6: Entry 12 of 14

File: USPT

Nov 23, 1999

DOCUMENT-IDENTIFIER: US 5991713 A

TITLE: Efficient method for compressing, storing, searching and transmitting  
natural language textApplication Filing Date (1):19971126Detailed Description Text (72):

Transmission of information between a remote server 12 and client computers 10 can be optimized by prefetching the hypertext links on a requested web page and transmitting their contents to the client computers 10 that are viewing the web page. When a user takes a link to a web page that has been prefetched, the web page already resides in a cache 84 on the client computer 10 and therefore, the browser 82 only needs to render it rather than to fetch it and render it. This reduces the time required to satisfy and render the browser's request for the linked web page.

[First Hit](#)   [Fwd Refs](#)☐ **Generate Collection**

L8: Entry 5 of 7

File: USPT

Aug 1, 2000

DOCUMENT-IDENTIFIER: US 6098064 A

TITLE: Prefetching and caching documents according to probability ranked need S  
listApplication Filing Date (1):  
19980522Detailed Description Text (43):

In an alternate embodiment, management of prefetching and caching of documents in the needs list is performed using a controlled aggressive policy. An example of a controlled aggressive policy is disclosed by Cao et al. in "Implementation and Performance of Integrated Application-Controlled File Caching, Prefetching, and Disk Scheduling," ACM Transactions on Computer Systems, 14, 311-343. In addition to the controlled aggressive policy, limited batch scheduling and two-level (or client-server) base management may also be adopted to manage the prefetching and caching of documents in the needs list. In limited batch scheduling, the client computer 102 issues batches of prefetch requests when the computer is idle to a server 104. Once the server 104, receives the batch request, the server delivers the documents in order of greatest need probability. Two-level base management allows client computers 102 to manage their own prefetching and caching, while the servers 104 define a global cache allocation policy.

[First Hit](#)   [Fwd Refs](#)

Generate Collection

L10: Entry 7 of 19

File: USPT

Jan 29, 2002

DOCUMENT-IDENTIFIER: US 6343323 B1

TITLE: Resource retrieval over a source network determined by checking a header of the requested resource for access restrictions

Application Filing Date (1):19981217Detailed Description Text (17):

After receiving and filling the requested identification or confirmation information, the client 1 sends the data and the cookie 19 back to the proxy 2. Using this information, the proxy 2 dispatches another billing request 20 to the ISB 5. After the ISB 5 acknowledges a successful billing, the proxy 2 attempts to locate the requested resource (or page) in its cache. If the resource is present in the cache, and the earlier retrieved header indicates that the resource remains valid, the proxy 2 delivers 24 the cached resource to the client 1. If the requested resource is not present in the cache, the original GET request is relayed 22 to the WWW Server 3. Then the WWW Server sends the requested resource to the proxy 2, where it is cached and forwarded 24 to the client 1. This ends the billing and access control transactions 25.

[First Hit](#)   [Fwd Refs](#)

Generate Collection

L10: Entry 18 of 19

File: USPT

Mar 7, 2000

DOCUMENT-IDENTIFIER: US 6035281 A

TITLE: System and method of multiparty billing for Web access

Application Filing Date (1):19970616Detailed Description Text (72):

If the cost for an access may be shared by multiple parties, it is preferable to indicate, on the object linked to the web page to be accessed, whether the user is responsible for payment of a portion of the access bill and, if yes, his/her share of the charge. This provides an incentive to users to access more web pages on the internet. This distinctive indication can be accomplished via creating different appearances of object linkages through coloring, special marks or graphical representations as explained below with respect to FIG. 3a. For example, an image of an advertisement can be modified to add the information about the billing parties and the modified image can be stored (cached) in disk device 8 at the OLSP proxy server 5. When the web page containing the advertisement is displayed by the web browser on a client computer, the advertisement image will be requested from the proxy server 5. The proxy server 5, upon receiving the request for the image, returns the locally cached modified image containing the billing information. In the preferred embodiment, the web page containing the advertisement image is not changed. Therefore, there is no requirement of prescanning or analyzing the web pages and altering their content. Only the advertisement image files are being changed. This can be achieved by substituting the image file with another one having the same URL, but with the billing information added. Since the modified image files are stored at the proxy server, the OLSP can manage the splitting of access charges among the user and any other participating parties based on the source web page and the destination (target) of the HTTP link associated with each advertisement image file.



First Hit

Generate Collection

L3: Entry 33 of 46

File: JPAB

Jul 2, 1999

~~PUB-NO: JP411175426A~~

DOCUMENT-IDENTIFIER: JP 11175426 A

TITLE: SERVICE REPEATER SYSTEM

~~PUB-DATE: July 2, 1999~~

## INVENTOR-INFORMATION:

NAME

IWATA, MASATAKE

COUNTRY

## ASSIGNEE-INFORMATION:

NAME

~~FUJII XEROX CO LTD~~

COUNTRY

APPL-NO: JP09341206

~~APPL-DATE: December 11, 1997~~INT-CL (IPC): G06 F 13/00; G06 F 12/00

## ABSTRACT:

PROBLEM TO BE SOLVED: To provide a service repeater system connected through a network or the like ~~between a service and a client~~ for operating relay between them in which a service including a screen display content with a large capacity from a service can be displayed and used even in a client whose memory capacity is limited.

SOLUTION: A content from a server 4 is divided by a data dividing part 20. The data dividing part 20 operates division according to the data receiving capacity of a portable information terminal 6 obtained from a terminal information part 22, and operates working and shaping so that each can have a format as an individual content document. For example, when division is operated in the middle of an instruction definition range, the data dividing part 20 restores the validity of an instruction by complementing an instruction tag. A data reconstituting part 24 adds the description of a button or the like to the content document so that a moving operation between the divided pages can be easily operated on the screen. The generated partial contents are held in a data storing part 26 and transmitted according to a request from the portable information terminal 6.

COPYRIGHT: (C) 1999, JPO

First Hit☐ Generate Collection

L3: Entry 41 of 46

File: DWPI

Jul 2, 1999

DERWENT-ACC-NO: 1999-434520

DERWENT-WEEK: 199937

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Service relay apparatus for server client system - has partial content generator that processes segment to generate partial content document with original content document format, such that generated partial content document is forwarded to client

PATENT-ASSIGNEE: FUJI XEROX CO LTD (XERF)

PRIORITY-DATA: 1997JP-0341206 (December 11, 1997)

Search Selected

Search ALL

Clear

## PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
<input type="checkbox"/> JP 11175426 A	July 2, 1999		012	G06F013/00

## APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
JP 11175426A	December 11, 1997	1997JP-0341206	

INT-CL (IPC): G06 F 12/00; G06 F 13/00

ABSTRACTED-PUB-NO: JP 11175426A  
BASIC-ABSTRACT:

NOVELTY - A partial content generator processes a segment to generate a partial content document with an original content document format. The generated partial content document is forwarded to a portable information terminal (6). A data dividing unit (20) divides the original content document into segments. A server accessing unit acquires the original content document from a server (4).

USE - For server client system.

ADVANTAGE - Enables utilizing the service provided by the server. Reduces necessary forwarding time since the size of the partial contents document is smaller than the original content. Reduces communication cost and network load since continuous changing of the communication circuit becomes unnecessary. Enables securing a user interface. Performs movement between pages quickly. DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the service relay apparatus and the block diagram of the server client system. (4) Server; (6) Portable information terminal; (20) Data dividing unit.

ABSTRACTED-PUB-NO: JP 11175426A  
EQUIVALENT-ABSTRACTS:

First Hit   Fwd Refs

Generate Collection

L8: Entry 6 of 7

File: USPT

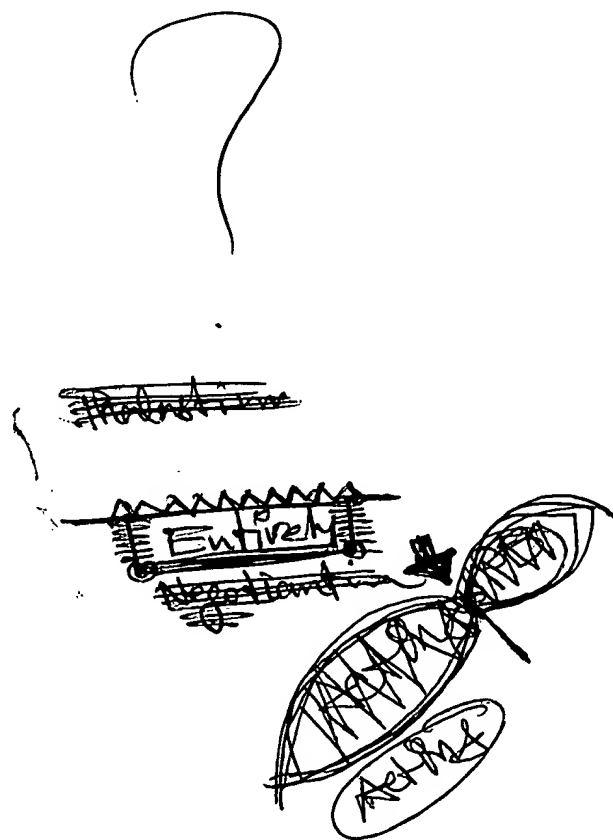
Jul 4, 2000

DOCUMENT-IDENTIFIER: US 6085193 A

TITLE: Method and system for dynamically prefetching information via a server hierarchy

Application Filing Date (1):  
19970929Detailed Description Text (94):

In the preferred embodiment of the present invention, a general prefetch policy for web and proxy servers has been described. Those skilled in the art will appreciate, however, that the present invention is applicable to any kind of situation where the objects to be prefetched have similar characteristics, and is not necessarily restricted to an Internet or WWW application.



[First Hit](#)   [Fwd Refs](#)

Generate Collection

Print

L3: Entry 8 of 46

File: USPT

Oct 30, 2001

DOCUMENT-IDENTIFIER: US 6311058 B1

**\*\* See image for Certificate of Correction \*\***

TITLE: System for delivering data content over a low bit rate transmission channel

Abstract Text (1):

The present invention provides a system by which information content data is delivered to a mobile device. The web content is divided into data and script information. The script information is used to operate on the data to render the data in a predetermined format.

Application Filing Date (1):19980630Brief Summary Text (10):

The present invention provides a system by which information content is delivered to a mobile device. The web content is divided into data and script information. The script information is used to operate on the data to render the data in a predetermined format.

[First Hit](#)   [Fwd Refs](#)

Generate Collection

Print

L3: Entry 10 of 46

File: USPT

Aug 14, 2001

DOCUMENT-IDENTIFIER: US 6275480 B1

TITLE: Fast associated control channel coding for speech channel

Application Filing Date (1):19971222Brief Summary Text (16):

On the other hand, if the mobile user terminal is currently in use, the logic leaves decision block 216 of FIG. 2b by the YES output, and flows to a further block 222. Block 222 represents the reading of the message to be transmitted to the selected mobile user terminal which is to be accessed. Block 224 represents the "addition" to the message of at least error detection codes, and preferably error detection and correction (EDAC) codes. The message is so encoded as to guarantee reception of the signals over a channel which is relatively weak. Unfortunately, the length of the message with error detection or EDAC codes is longer than the slot interval. Consequently, the logic flows to a block 226, which represents the division of the message into four equal-content portions, each of which is smaller than a slot or burst, meaning that the data content of each of the four portions can be fitted into a single slot interval. It should be understood that the number four is appropriate for a particular embodiment of the invention described in the abovementioned Schiavoni patent application, and may differ for other slot intervals or message lengths, or for error coding of different robustness. Block 228 represents "addition" of error correction codes to each of the four equal content portions of the message. Thus, the complete message is encoded a first time with at least error detection codes, and after division into four (or any number) of portions, the individual equal-content portions of the message are each again encoded, this time with error correction codes. Error correction codes are added to each equal-content portion of the original coded message, in an amount sufficient to fill a slot interval or slot data-carrying ability. In one mode of operation of the flow chart of FIG. 2b, the logic flows to a further block 229, which represents the interleaving of the bits of each of the equal-content error-coded portions of the message, to distribute errors throughout the slot, and also to aid in identifying a call maintenance message, as described below. Whether the bits are interleaved or not, the logic flows from block 228 to a further block 230, which represents the extraction, from the log previously used, of the slot set in which the mobile user terminal is using. Finally, the logic flows by way of a logic path 232 to a further block 234, which represents substitution of the four equal-content error-coded messages into the next four slots of traffic data allocated to the mobile user terminal in question, as described above in relation to FIG. 2a. The logic then ends at block 220.

First Hit☐ Generate Collection

L3: Entry 35 of 46

File: EPAB

Jul 15, 1999

PUB-NO: WO009935802A2

DOCUMENT-IDENTIFIER: WO 9935802 A2

TITLE: SYSTEM FOR DELIVERING DATA CONTENT OVER A LOW BIT RATE TRANSMISSION CHANNEL

PUBN-DATE: July 15, 1999

## INVENTOR-INFORMATION:

NAME

COUNTRY

WECKER, DAVE

DEO, VINAY

MILLER, JOHN MARK

TUNIMAN, DAVID

O'LEARY, MICHAEL J

## ASSIGNEE-INFORMATION:

NAME

COUNTRY

MICROSOFT CORP

US

APPL-NO: US09900336

APPL-DATE: January 7, 1999

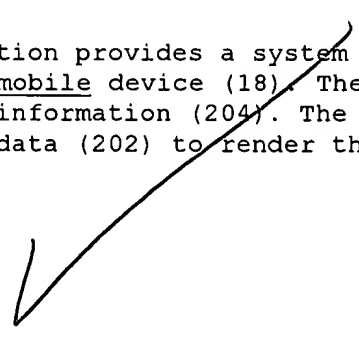
PRIORITY-DATA: US07072098P (January 7, 1998), US07512398P (February 13, 1998),  
US10766698A (June 30, 1998)

INT-CL (IPC): H04 L 29/06

EUR-CL (EPC): G06F001/32; G06F017/21, G06F017/22 , H04L012/28 , H04L012/56 ,  
H04L029/06 , H04L029/06 , H04L029/06 , H04L029/12 , H04Q007/14

## ABSTRACT:

CHG DATE=19990803 STATUS=O>The present invention provides a system by which information content (250) is delivered to a mobile device (18). The web content (250) is divided into data (202) and script information (204). The script information (204) is used to operate on the data (202) to render the data (202) in a predetermined format.



First Hit

Generate Collection

L1: Entry 3 of 6


File: EPAB

Jun 13, 2001

DOCUMENT-IDENTIFIER: EP 1107539 A2

TITLE: Apparatus for supplying content to a portable terminal

Abstract Text (1):

CHG DATE=20010704 STATUS=O> Between a portable terminal 2 and a content server there is interposed a divided content supply section, in order to obtain the content demanded by the portable terminal from the content server and transmit it to the portable terminal to avoid useless content prefetching at a gateway server. The divided content supply section 14 transmits the content demanded by the portable terminal 2 to obtain by divided content unit each amount of information that can be displayed by the portable terminal 2. A prefetching section 17 prefetches the other contents referred to from the divided content, by a divided content unit transmitted to the portable terminal 2 from a, or a group of content servers 3 and stores it in a content hold section 11. 

*Same*

First Hit   Fwd Refs☐ **Generate Collection**

L10: Entry 9 of 19

File: USPT

Aug 7, 2001

DOCUMENT-IDENTIFIER: US 6272598 B1

TITLE: Web cache performance by applying different replacement policies to the web cache

Application Filing Date (1):  
19990322Brief Summary Text (8):

Ideally, the cache at the proxy server 16 retains all of the cached objects. However, the typical storage capacity of the proxy server 16 is in the range of 256 megabytes to 1 terabyte, with most Web proxy capacity being at the lower half of the range. Therefore, it is important to form a replacement strategy or policy for determining which objects are to be evicted or replaced from the cache of the proxy server 16 when a recently received object is to be cached within exhausted storage space. Two important metrics that are used to measure proxy cache performance are cache hit rate and byte hit rate. The cache hit rate is the percentage of all user requests that are satisfied by the proxy server 16, rather than by access to the original server 18. The byte hit rate is the percentage of all network traffic, measured in bytes, transferred directly from the proxy server 16, instead of across the external network.

Brief Summary Text (9):

There are a number of cache replacement strategies or policies that have been proposed for Web proxy caching. However, the cache in the prior art proxy server typically uses only one replacement policy to manage the content of the cache. As is known, each replacement policy has its advantages and disadvantages, and is suitable for one kind of objects. This means that each of the replacement policies improves the performance of the cache in the proxy server for a chosen metric (e.g., hit rate, byte hit rate, or latency). For example, a replacement policy of evicting the largest document (rather than a number of smaller documents) may improve cache hit rate, but does not improve the byte hit rate. As a further example, a replacement policy of evicting the smallest document may increase the byte hit rate, but does not improve the cache hit rate.

Detailed Description Text (15):

There are a number of replacement policies or strategies that can be used for proxy caching. However, each of the policies is optimized for one chosen metric (e.g., hit rate, byte hit rate, or latency) and has its own advantages and disadvantages. For example, a well known replacement policy is the Least-Recently-Used (LRU) algorithm. According to this policy, when an eviction is required in order to store a recently received object, the previously cached object that was requested least recently is evicted. Although this policy is a simple policy that is effective for virtual memory systems, it does not work well if the cached objects are only popular during certain times of a day or certain days of a month.